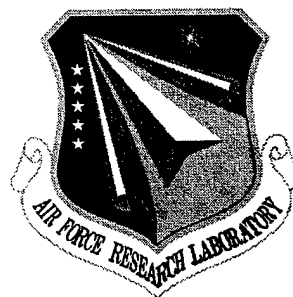


AFRL-IF-RS-TR-2001-102
Final Technical Report
May 2001



OPEN TECHNOLOGY FOR SOFTWARE EVOLUTION: HYPERWARE, ARCHITECTURE, AND PROCESS

The Regents of the University of California

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. E102

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

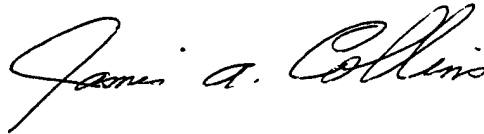
20010713 044

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2001-102 has been reviewed and is approved for publication.



APPROVED: JAMES R. MILLIGAN
Project Engineer



FOR THE DIRECTOR: JAMES A. COLLINS, Acting Chief
Information Technology Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFTD, 525 Brooks Road, Rome, NY 13441-4505. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

OPEN TECHNOLOGY FOR SOFTWARE EVOLUTION: HYPERWARE,
ARCHITECTURE, AND PROCESS

Richard N. Taylor
David Redmiles

Contractor: The Regents of the University of California
Contract Number: F30602-97-2-0021
Effective Date of Contract: 1 November 1996
Contract Expiration Date: 30 April 2000
Short Title of Work: Open Technology for Software Evolution:
Hyperware, Architecture, and Process
Period of Work Covered: Nov 96 – Apr 00

Principal Investigator: Richard N. Taylor
Phone: (942) 824-6429
AFRL Project Engineer: James R. Milligan
Phone: (315) 330-3013

Approved for public release; distribution unlimited.

This research was supported by the Defense Advanced Research
Projects Agency of the Department of Defense and was monitored
by James R. Milligan, AFRL/IFTD, 525 Brooks Rd, Rome, NY.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE May 01		3. REPORT TYPE AND DATES COVERED Final Nov 96 - Apr 00
4. TITLE AND SUBTITLE OPEN TECHNOLOGY FOR SOFTWARE EVOLUTION: HYPERWARE, ARCHITECTURE, AND PROCESS			5. FUNDING NUMBERS C - F30602-97-2-0021 PE - 62301E PR - E102 TA - 01 WU - 01	
6. AUTHOR(S) Richard N. Taylor and David Redmiles				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) The Regents of the University of California University of California, Irvine Irvine CA 92717-1875			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency 3701 North Fairfax Drive Arlington VA 22203-1714			10. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-2001-102	
11. SUPPLEMENTARY NOTES AFRL Project Engineer: James R. Milligan, IFTD, 315-330-3013				
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This effort, sponsored by DARPA's evolutionary Design of Complex Software (EDCS) Program with additional funding and support from the Air Force Research Laboratory, has focused on several technologies to aid in the effective evolution of complex software-intensive systems, particularly: Open, broad-based hyperprogram and active rationale support; Component-based architecture evolution technology; and Multi-stakeholder process support. Open hyperware, software architecture, and workflow/process technologies are key enablers for supporting the evolution of complex systems, and this effort developed techniques, standards, and tools in each of the above areas.				
14. SUBJECT TERMS Evolution, software, architecture, process, hyperware, www, design workflow			15. NUMBER OF PAGES 40	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

TABLE OF CONTENTS

<u>SECTION</u>	<u>TITLE</u>	<u>PAGE</u>
1	Accomplishments	1
1.1	Hyperware	2
1.1.1	HTTP/URI/Apache	2
1.1.2	WebDAV	3
1.1.3	Additional Web Research	4
1.1.4	Chimera	5
1.2	Software Architecture	6
1.2.1	C2 Architectural Style	6
1.2.2	Argo/UML Design Environment	8
1.2.3	Dynamic Architectures/ArchStudio	9
1.2.4	Event-based Architectures	11
1.2.5	C2 and Java Beans	11
1.3	Process	12
1.3.1	Endeavors	12
1.3.2	EDEM	15
1.3.3	Knowledge Depot	16
2	Professional Personnel	17
3	Publications in Technical Journals	19
4	Final Resource Status Report	20
5	References	21

LIST OF TABLES

<u>NUMBER</u>	<u>TITLE</u>	<u>PAGE</u>
1	Contract Resource Status: 11/01/96 - 04/30/2000	20

1. Accomplishments

Our work has focused on several technologies to aid in the effective evolution of complex software. In particular our research foci have been:

- open, broad-based hyperprogram and active rationale support
- component-based architecture evolution technology, and
- multi-stakeholder process support

Open hyperware, software architecture, and workflow/process technologies are key enablers for supporting evolution of complex software systems. We have developed techniques, standards, and tools in each these areas and have distributed these results via the World-Wide Web (WWW). The technologies have been developed cooperatively, allowing results in one area to leverage the other two. The hyperware focus has been on developing infrastructure to integrate open link server hypermedia systems with the WWW, while adding support for hyperweb configuration management, semantically-rich link generation, cooperation agents, and automatic rationale capture. The architecture research has been directed at reducing the cost of application development by providing domain-informed, style-based design guidance, architecture design environments with analysis capabilities, implementation technologies, and support for dynamic change. The process work has been designed to improve coordination and managerial control of teams by allowing flexible definition, modeling, and execution of typical workflow and process applications over the Internet.

Highlights of our accomplishments under this contract include:

- The Apache Web Server has been selected for the 1999 ACM Software System Award. This highly distinguished honor is awarded to an institution or individual(s) recognized for developing a software system that has had a lasting influence, reflected in contributions to concepts, in commercial acceptance, or both. Past recipients include Unix, Interlisp, SmallTalk, Postscript, TCP/IP, and WWW/Mosaic. Roy Fielding, as co-founder and current chairman of the Apache Software Foundation will be accepting the award.
- The Internet Engineering Task Force (IETF) approved the URI (RFC 2396) [BFM98], HTTP/1.1 (RFC 2616) [FGM+99], and WebDAV (RFC 2518) [GWF+99] specifications as proposed standards. This indicates that these standards are stable, of high technical quality, and have undergone significant community review. UC Irvine EDCS researchers were major contributors to all three standards efforts.
- We have seen wide scale adoption of the UC Irvine led WebDAV protocol (RFC 2518) [GWF+99] for collaborative authoring of Web content. Adopters include Microsoft (Office 2000, IE5, IIS 5), IBM (DAV4J), and open source projects (Apache mod_dav, DAV Explorer). This has resulted in 10 client applications, 2 publicly available client API libraries, 17 WebDAV servers (12 class-2, 5 class-1), and 2 WebDAV-enabled Web storage sites. These applications run on a range of platforms, with client support for Windows, Unix, and MacOS, and server support for Windows and Unix.
- Northrop-Grumman Military Aircraft Systems Division successfully used Chimera for the creation of a hyperweb of software development artifacts containing hundreds of thousands of hypermedia entities.
- An Internet start-up company, ETI, has licensed Endeavors web-based process/workflow technology for use as a commercial software system. Endeavors, created by EDCS researchers at UC Irvine, enables customizable definition and execution of distributed workflow models involving multiple activities and resources. ETI is actively developing this technology and is supplying workflow solutions to commercial customers.

- UC Irvine licensed the JavaBrain technology to Sun Microsystems, Inc. Sun is using the JavaBrain technology for the development and deployment of Computer Based Training (CBT) materials. JavaBrain is built on top of the Endeavors technology.
- Northrop-Grumman Military Aircraft Systems Division successfully used the C2-style and implementation framework for the modeling and implementation of a B-2 avionics simulation environment.
- Argo/UML, an open source architecture design environment utilizing the industry standard Unified Modeling Language (UML), has been widely used by over 19,000 registered users and by over 150 developers at sites such as Aonix, ISX, IBM, Motorola, and Oracle. Argo/UML has been integrated into the TIGRIS open-source development environment <<http://argouml.tigris.org/>>
- Researchers at UC Irvine have constructed ArchStudio 2.0, an extensible, integrated software architecture development environment. ArchStudio 2.0 integrates several existing EDCS technologies including Argo/C2, ArchShell, DRADEL, and Chimera. ArchStudio 2.0 uses an XML-based abstract model of the system architecture as a basis for integration. The defined XML DTD, called xADL, is designed to support a range of architectural models, including C2.
- Our researchers and staff organized the first Workshop on Internet-Scale Event Notification (WISSEN). Attendees included The Aerospace Corp, Microsoft, Tibco, BLIP, Lotus, Oracle, Sun Microsystems, Netscape, Activeverse, Novel, and Hewlett Packard, Raytheon, GTE, Lucent, IBM Research, FileNet, and Tandem.
- Our researchers and staff organized the 1999 Workshop on Internet-scale Software Technologies (TWIST 99) - a follow-on to the first Workshop on Internet-Scale Event Notification (WISSEN). The workshop was held August 19-20 at UC Irvine. Participants included The Aerospace Corporation, Lotus Development, Intel Labs, MIT, Network Solutions, Xerox Palo Alto Research Center, Information Sciences Institute/USC, and CNRI.
- Research supported by this contract has led to 12 technical journal publications, 50 conference and workshop publications, 3 IETF standard protocols, and 6 doctoral dissertations.

1.1. Hyperware

Our hypermedia work represents two different approaches to providing hyperware capabilities. Chimera embodies a link server approach while the WWW embodies a distributed link data approach. A link server provides greater control and coherency management of the hypermedia relationships and dependencies; distributed link data provides for better scalability and hyperweb robustness. We believe that the advantages of both approaches are necessary for a hyperware environment to support the complete evolution of complex software systems. We have used our experience with both approaches to provide the best combination of capabilities. In addition, our leadership within the WWW standardization efforts (e.g. HTTP, URI, and WebDAV standards) has provided us with the unique ability to transition this work into the next generation of World Wide Web technology.

1.1.1. HTTP/URI/Apache

In June, 1999, the IESG published our specification of the Hypertext Transfer Protocol (HTTP/1.1) as a Draft Standard (RFC 2616) [FGM+99]. HTTP/1.1 is an application-level protocol for distributed, collaborative, hypermedia information systems and is the primary transfer protocol for the World Wide Web. It is a generic, stateless protocol that can be used for many tasks beyond its use in hypertext, such as name servers, distributed object management systems, and global software engineering environments, through extension of its request methods, error codes and

header fields. A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred.

[<http://www.ics.uci.edu/pub/ietf/http/>](http://www.ics.uci.edu/pub/ietf/http/)

We have assisted in the implementation of HTTP/1.1 within the Apache server (Fielding is one of the original Apache developers). Apache remains the market-leading web server technology. The Apache Web Server has been selected for the 1999 ACM Software System Award [<http://www.acm.org/awards/ssaward.html>](http://www.acm.org/awards/ssaward.html). This highly distinguished honor is awarded to an institution or individual(s) recognized for developing a software system that has had a lasting influence, reflected in contributions to concepts, in commercial acceptance, or both. Past recipients include Unix, Interlisp, SmallTalk, Postscript, TCP/IP, and WWW/Mosaic. The Software System Award carries a prize of \$10,000 which will go to the Apache Software Foundation. Roy Fielding, as co-founder and current chairman of the Apache Software Foundation will be accepting the award.

[<http://www.apache.org/>](http://www.apache.org/)

Roy Fielding has also authored the specification of Uniform Resource Identifiers (URI) [BFM98], the technology for identifying resources on the Internet, was accepted by the Internet Engineering Steering Group (IESG) for publication as Draft Standard RFC 2396.

[<http://www.ics.uci.edu/pub/ietf/uri/>](http://www.ics.uci.edu/pub/ietf/uri/)

1.1.2. WebDAV

The World Wide Web was originally conceived as an interactive, read/write medium, and it is only the early dominance of the Mosaic browser which cast the Web into its current publish/browse paradigm. While the HyperText Transfer Protocol (HTTP/1.1) contains limited support for remote authoring in addition to remote browsing of content, this support is not sufficient for collaborative authoring of web content. Jim Whitehead is the Chair of the Internet Engineering Task Force World Wide Web Distributed Authoring and Versioning (WEBDAV) Working Group, which is developing interoperability specifications for how to extend HTTP to perform remote authoring and versioning of web content. WebDAV provides a network protocol for creating interoperable, collaborative applications. Major features of the protocol include locking (concurrency control), XML properties, and namespace manipulation [Whi97,Rad97,Dud98,Han98,Wal98].

The WebDAV Working Group, led by Jim Whitehead, has obtained Internet Engineering Task Force (IETF) approval of the WebDAV Distributed Authoring Protocol as a Proposed Standard (RFC 2518, [GWF+99]). This indicates the WebDAV standard is stable and of high technical quality, has undergone significant community review, and is generally believed to have resolved known design choices. Approval by the IETF also acts as a signal for corporations and other organizations to begin adopting the protocol in their products. We have seen wide scale adoption of the WebDAV for collaborative authoring of Web content. Adopters include Microsoft (Office 2000, IE5, IIS 5), IBM (DAV4J), and open source projects (Apache mod_dav, DAV Explorer). This has resulted in 10 client applications, 2 publicly available client API libraries, 17 WebDAV servers (12 class-2, 5 class-1), and 2 WebDAV-enabled Web storage sites. These applications run on a range of platforms, with client support for Windows, Unix, and MacOS, and server support for Windows and Unix. Together, these developments show the benefits of EDCS involvement in network protocol standardization, since once the standard has been approved, significant development coalesces around the standard, leveraging DARPA investments. Some examples of WebDAV adoption include:

- Wide industry support for WebDAV has included client support of WebDAV on Microsoft's Office 2000 release, and server support on Microsoft's Windows 2000 operating system. Microsoft's Internet Explorer web browser, version 5, also shipped with WebDAV support.
- IBM has likewise supported WebDAV through DAV4J, a freely available WebDAV server, and Java client API for WebDAV, available on IBM's AlphaWorks site.
- A WebDAV module for the Apache Web server (the module is called "mod_dav") was released in beta form, as part of ongoing development. Red Hat announced that they are bundling the mod_dav WebDAV support module for Apache in with their Apache Commerce Server product.
- Zope, the "Z Object Publishing Environment" (<http://www.zope.org>), a product of Digital Creations, also announced their support for WebDAV during the reporting period.
- Sharemation <<http://www.sharemation.com/>> announced the availability of a WebDAV-compliant storage site, where it is possible for anyone to sign up for free 20 Meg of storage in under 5 minutes. This was a big step forward for WebDAV, since it means that anyone can quickly get access to a WebDAV server to try out the protocol.
- The first Macintosh WebDAV client, called Goliath, was announced on September 11th. It provides a Finder-like interface to a remote WebDAV server.

The WebDAV project at UC Irvine released the first WebDAV client, DAVExplorer. DAVExplorer was released open source allowing for a wide pool of developers to study the WebDAV protocol and to improve the design of DAVExplorer. DAVExplorer has successfully been used as a test bed for WebDAV servers from Apache, Microsoft, and IBM. We have continued developing WebDAV Explorer to make it fully compliant with the final WebDAV specification, RFC 2518, and to upgrade it to use the Java JDK 1.2 libraries.

Our work on WebDAV has continued with progress on the Advanced Collections, Versioning and Configuration Management (Delta-V), and Searching (DASL) protocols. UCI/EDCS was instrumental in helping form a follow-on effort to WebDAV called DASL, DAV Searching and Locating. The DASL effort, currently in-process of becoming an IETF working group, will address issues of how to remotely search a repository which contains a set of Web DAV resources. Alex Hopmann, Microsoft, is chair of the DASL group. Versioning

Concerns for WebDAV's use of the Extensible Markup Language's (XML) MIME media type "text/xml" were addressed. This has led to work on *XML Media Types* [WM98], a document which registers the "text/xml" and "application/xml" media types. During the reporting period, this document went through 6 major revisions, and received significant review from the World Wide Web Consortium's XML Special Interest Group and the IETF Media Types community. This document was submitted to the IESG and is currently awaiting approval.

<<http://www.ics.uci.edu/pub/ietf/webdav>>

<<http://www.webdav.org>>

1.1.3. Additional Web Research

Rohit Khare has focused on tracking development of HTTP-NG and researching the potential for third-party extensibility using Web proxy servers, Extensible Markup Language, and the current generation of HTTP. He was author of *Seventh Heaven*, a bimonthly column on application-layer protocol design in IEEE Internet Computing [Kha 98, Kha98-2, Kha98-3]. Topics covered include the contributions of Jon Postel to protocol specification, and histories of Network News Transfer Protocol (NNTP), Gopher, and Hypertext Transfer Protocol (HTTP).

Roy Fielding completed a survey of software architectural styles that characterizes styles in terms of their impact on system communication across a network. This research is to provides a design framework to help software architects and application-level protocol designers choose an appropriate architectural style for the particular communications characteristics of their application. [FT00]

1.1.4. Chimera

We complete work on Chimera 2.0. This all-Java release is deeply integrated with the WWW and provides distributed access to hypermedia information. Chimera 2.0 clients can be either stand-alone applications or Java applets. URLs are used to reference Chimera hyperwebs and HTTP is used to establish connections between clients and servers. Java and C APIs are available for Chimera client integration. Integrated clients include Xemacs, Netscape 3.0, FrameMaker 5.0, JPEG and GIF viewers. The latest release includes the ability to import and export hyperwebs in XML (Extensible Markup Language) format, and the ability to select among a choice of user-interface look-and-feels.

Chimera was used successfully by two aerospace companies, Northrop-Grumman Military Aircraft Systems Division and Raytheon. Northrop Grumman was able to take advantage of Chimera's XML import feature to automatically process large documents into Chimera databases and generate hyperwebs containing hundreds of thousands of hypermedia entities. A Chimera installation was achieved at Raytheon for a EDCS2 grant proposal feasibility study. Raytheon used Chimera to access various artifacts used on the F-15 project.

Scalability issues arose in supporting the large number of hypermedia entries generated by Northrop-Grumman. In response, we improved Chimera's support for scalability on data intensive operations by two orders of magnitude, from thousands to hundreds of thousands of entities handled. This was accomplished by adopting the use of a freeware relational database (MYSQL) and developing the capability to use filtering mechanisms over a large scale hyperweb. The use of MYSQL enabled us to respond to Northrop-Grumman's demands in a rapid fashion (the increase of Chimera's scalability occurred in less than four weeks). This work was published in [And99,And99-2].

Later improvements to Chiron 2 included:

- improved XML support
- A selectable traversal algorithm function which will allow client applications and users to select the traversal behaviors for Chimera. This allows the user to apply rules to traversals across the n-ary links Chimera supports, such as a rule to traverse to documents ordered with the newest linked document appearing first
- a new enhancement known as anchor and link filtering, which allows users to restrict links made visible on the Chimera Server's display to those of a specific type
- A Chimera COM API on the Win32 platform. Chimera's API is now available as a COM object that can be used to integrate Win32 applications. These integrations involve an application using a DDE interface to the COM API which then accesses Chimera
- A client integration of Adobe's Frame Maker using the COM API
- A port of the Chimera to the NT platform

The integration of Chimera with the Rivendell Tool Server, developed by Gail Kaiser's group at

Columbia University, was successfully completed. Further refinement of the integration was achieved with feedback provided for future versions of Rivendell. Columbia University replaced Xanth with Chimera as the underlying hypermedia infrastructure on Oz web.

Additional work on the Chimera open hypertext project has involved porting Chimera to other platforms as well as supporting users experimenting with ports to other systems. Our work included refining our Linux port and tests of Chimera 2.0 on Windows 95, Windows 98, and Windows NT. From these tests, it was discovered that both Windows 95 and Windows 98 have incomplete Java Virtual Machine implementation available which limited the ability of Chimera 2.0 to be ported successfully onto these platforms. The tests have lead to further refinement of Chimera's servers and client applications on Linux and Windows NT.

In support of cross platform activities, a new client integration was completed and tested in-house. for the Adobe's Acrobat Reader for Unix based systems. The integration was completed using the plug-in capability of the software.

Further Chimera research was undertaken at the University of Colorado at Boulder. The focus of work at CU was on the refinement of the servers and data models for Chimera. This has resulted in an new publicly available Chimera version, Chimera 3.0. Current work has focused on implementing first-class typing of anchors and links within the Chimera hypermedia model.

[<http://www.ics.uci.edu/pub/chimera/>](http://www.ics.uci.edu/pub/chimera/)

1.2. Software Architecture

Software architecture is a key enabler for evolutionary development of complex systems. Our approach to architecture is based on notions of style, composition, dynamic change, visualization, analysis, system generation, and active rationale support. Each of these elements has direct bearing on evolvability of a system. Architectural style is an abstraction of recurring composition and communication characteristics of a set of architectures. As such, styles are key design idioms that enable exploitation of suitable structural and evolution patterns and facilitate component and process reuse. Composition of pre-existing software components reduces time to market. Dynamic change supports in-field evolution of systems can cannot afford to be brought down. Visualization of system architectures facilitates understandability by all stakeholders in a software development process. Analysis at the architecture level makes predicting overall system behavior tenable. System generation facilities automate the task of building each system version and, thus reduce the cost of evolution. Finally, an architecture-based framework for design rationale not only facilitates rationale capture but also provides a framework for effective delivery and analysis of design rationale, so that designers get the information they need, when they need it.

1.2.1. C2 Architectural Style

We have designed and implemented a framework of abstract classes for modeling C2-style architectures [TMA+96] with concepts, such as components, connectors, communication ports, and so on. Among other things, the framework enables users to manually wrap OTS components. We have demonstrated this with two different implementations of the framework: one in C++ and the other in Java. Sources for both frameworks and demonstration applications are available on the WWW.

[<http://www.ics.uci.edu/pub/arch/c2.html>](http://www.ics.uci.edu/pub/arch/c2.html)

We have devised a comparison and classification framework for architecture description languages (ADLs) and performed an extensive survey based on that framework. This survey highlights existing ADL capabilities and shortcomings, and aided us in our investigation of an ADL for C2-style architectures [MT99].

Software architectures have the potential to substantially improve the development and evolution of large, complex, multi-lingual, multi-platform, long-running systems. However, in order to achieve this potential, specific architecture-based modeling, analysis, and evolution techniques must be provided. One aspect of our current research in software architectures focuses on a type theory for software architectures, which allows flexible, controlled evolution of software components in a manner that preserves the desired architectural relationships and properties. Critical to the type theory is a taxonomy that divides the space of subtyping relationships into a small set of well defined categories. In the context of this work, we are investigating the effects of large-scale development and off-the-shelf reuse on establishing type conformance between interoperating components in an architecture [MRT98]. Furthermore, one cannot fully benefit from such techniques unless support for mapping an architecture to an implementation also exists [MRT99].

One aspect of the research conducted this reporting period is the construction of an architecture-based modeling, development, and evolution toolsuite, called DRADEL. DRADEL is an outgrowth of our experience with systems developed and evolved according to the C2 architectural style. We formalized the syntax and semantics of an architecture description language (ADL) specifically designed to support architecture-based evolution and enumerated the kinds of evolution the language supports. We used the ADL as the basis of DRADEL, a component-based environment that enables modeling, analysis, and evolution of architectures expressed in the ADL, as well as mapping of architectural models to our implementation infrastructure (the C2 class framework). The architecture of the DRADEL environment itself can be evolved easily to support multiple ADLs, kinds of analyses, architectural styles, and implementation platforms. Our approach is fully reflexive: DRADEL can be used to describe, analyze, evolve, and (partially) implement itself, using the very ADL it supports [MRT99].

[<http://www.ics.uci.edu/pub/arch/ADL/SADL.html>](http://www.ics.uci.edu/pub/arch/ADL/SADL.html)

Another aspect of our architectural research deals with implementing complex software connectors. Since architecture-level components often contain complex functionality, it is reasonable to expect that their interactions will also be complex. Modeling and implementing software connectors thus becomes a key aspect of architecture-based development. Software interconnection and middleware technologies such as RMI, CORBA, ILU, and ActiveX provide a valuable service in building applications from components. The relation of such services to software connectors in the context of software architectures, however, is not well understood. To understand the trade-offs among these technologies with respect to architectures, we have evaluated several off-the-shelf middleware technologies and identified key techniques for utilizing them in implementing software connectors. We have integrated four such technologies with the C2 implementation infrastructure: University of Colorado's Q system, Sun's Java RMI facility, University of Maryland's Polyolith software bus, and Xerox PARC's ILU distributed object system. Our platform for investigation was the C2 style. By encapsulating middleware functionality within software connectors, we have enabled the coupling of C2's existing benefits such as component interchangeability, substrate independence and structural guidance with new capabilities of multi-lingual, multi-process and distributed application development in a manner that is transparent to architects

[MDT98, DMT99].

1.2.2. Argo/UML Design Environment

JArgo, a prototype Java-based interactive software architecture design environment was completed. JArgo allows for graphical visualization and direct manipulation of architectural models. Architectural components and their relationships are modeled as connected graphs and presented via multiple, coordinated, customizable views. The architect can build, modify and analyze the architecture via Argo. The environment supports the design and dynamic manipulation of applications in the C2-style, "critic"-based analyses of architectural designs, and initial rationale capture agents. Graph editing support in Argo is provided by the Graph Editing Framework (GEF). GEF is a library of Java classes that make it easier to develop new applications that involve diagram editing and connected graph editing.

[<http://www.ics.uci.edu/pub/arch/argo/>](http://www.ics.uci.edu/pub/arch/argo/)

[<http://www.ics.uci.edu/pub/arch/gef/>](http://www.ics.uci.edu/pub/arch/gef/)

We have investigated the use of UML (the Unified Modeling Language) to represent software architectures. UML is a object-oriented design notation that has recently be standardized by the Object Management Group. UML has wide-spread industry interest and growing tool support. Unlike previous object-oriented design notations, UML is formally defined and provides a constraint language and extension mechanisms. We have been able to use the UML constraint language to express most of the guidelines of the C2 style and some aspects of the Wright architecture description language [RMRR98]. Expressing software architecture concepts in a standard design language is an important step toward wide-spread use of software architecture models.

We have revised the design and implementation of the Argo software architecture design environment to make it more efficient and flexible, and to ease integration with design tools implemented in Java. Specifically, we have implemented versions of the UML meta-model and associated visualizations in GEF and critics in Argo. Argo/UML is an object-oriented design tool that has cognitive support features as found in the existing Argo/C2 tool supporting C2-style software architectures. Like the previous Argo/C2 tool, Argo/UML has design critics and a dynamic "to do" list that together help designers resolve identified problems in a design. We have done a significant amount of development on Argo/UML resulting in several beta releases. New features include:

- improved scalability in the critiquing system
- a new XML-based file format
- new design visualization features
- customizable navigational perspectives that structure the design around specific design tasks
- checklists which save time in later design reviews
- novel diagram editing tools
- support for attaching problem-fixing wizards to critics
- new support for UML activity diagrams
- table views of the design
- enhanced "clarifier" icons that visually highlight problems
- and a cooperative design query mechanism.

Argo/UML has been widely used by over 19,000 registered users and by over 150 developers at sites such as Aonix, ISX, IBM, Motorola, and Oracle. Both Argo and GEF are currently being integrated into the TIGRIS open source development platform.

[<http://argouml.tigris.org/>](http://argouml.tigris.org/)

[<http://gef.tigris.org/>](http://gef.tigris.org/)

In addition, a survey of design critiquing systems is was completed [Rob98].

We conducted a user study of the “broom alignment tool” feature of Argo/UML. The study consisted of ten subjects who performed sample diagramming tasks using the broom alignment tool or standard alignment commands. The results of this study showed that the broom alignment tool required much less mouse movement and dragging, that users enjoyed using the broom tool, and that the short-term memory load required to use the broom was not significantly different from that of standard tools. The results of this study were published in RKR99].

Jaya Vaidyanathan and Jason Robbins explored the use of HTML as a user interface prototyping medium. HTML is an attractive prototyping medium because many, high quality HTML editors are available and because the resulting prototypes emphasize elements of the design that is hard to express with paper prototypes, also HTML prototypes can be evaluated by remote subjects. An example user interface was prototyped and a prototyping method was outlined. The results of this study were published in [VRR99].

1.2.3. Dynamic Architectures/ArchStudio

We have conducted research on techniques for supporting dynamic modification of software architectures to enable runtime evolution of mission- and safety-critical systems. We have used the C2 Java framework as a basis for a prototype tool, *ArchShell*, which enables users to interactively modify architectures after system generation. ArchShell can currently also perform some modifications of architectures at runtime. An initial version of this tool has been released. ArchShell was then integrated with the *JArgo* architecture design environment using a tool integration strategy based on the C2-style. The integrated environment, called *ArchStudio*, enables architects to design a system’s architecture using JArgo’s graphical user interface and analysis facilities, and subsequently execute the system using ArchShell. While the system is executing, the architect may continue to modify the system’s architectural model. These modifications are instantly reflected in the running system using ArchShell. An external tool was then added to ArchStudio that enables new components to be downloaded from a Web browser and automatically installed into a running system.

[<http://www.ics.uci.edu/pub/arch/dynamic-arch.shtml>](http://www.ics.uci.edu/pub/arch/dynamic-arch.shtml)

Notable progress was made on understanding how software architectures could be used as a basis for runtime software evolution. The contribution of architectural style and software connectors was explored and described in a paper presented at the International Conference on Configurable Distributed Systems [OT98]. We have been studying the role of connectors in supporting the modeling, analysis, and execution of software systems. In particular, the beneficial role of connectors in flexibility, heterogeneity, code mobility, and distribution are described in [ORT98].

Peyman Oreizy completed a survey that compares different approaches to Decentralized Software Evolution (DSE). DSE enables third-parties to evolve a software application independent of the organization that originally developed it. Application vendors employ DSE as a means of attract-

ing additional users to their applications - and, consequentially, increasing their market share. This benefits everyone involved: the original application vendor sells more product since customization constitutes use; third-party developers deliver a product in less time and with lower cost by reusing software as opposed to building it from scratch; and customers receive a higher quality product, customized to suit their needs, in less time and with lower cost. While there are a variety of approaches to DSE, there were no clear characterizations of their differences or of their relative benefits and shortcoming. These techniques include application programming interfaces (APIs), scripting languages, software plug-ins, etc. The survey covers different approaches to DSE and helps answer the following questions: What is decentralized software evolution and why is it important? What are the characteristic approaches to DSE? What are the important issues to consider when evaluating different DSE approaches? How do different approaches compare and what are their benefits and shortcomings? Where do current approaches fall short? What are some open areas for further research and development? The results of this survey can be found in [Ore98, Ore98-3].

The C2 group, performed an integration of ArchStudio with the Armani constraint language from CMU (PI: David Garlan, contact: Robert Monroe). The integration enables architects to specify constraints on how a software architecture can evolve during runtime. As runtime software changes are requested, ArchStudio queries Armani to determine whether or not the change would leave the system in a consistent configuration. If the change violates a constraint, the change request is rejected; otherwise the change is applied to the running system by ArchStudio. This capability provides additional assurances that runtime software upgrades do not violate application integrity. The integration also enables ArchStudio to generate ACME architectural descriptions.

We had a successful collaboration with Northrop Grumman (PI: Greg Johnson) on the use of a C2-style architecture in modeling and implementing the B-2 avionics simulation environment. Northrop Grumman successfully used ArchStudio/C2 as a tool integration framework for the B-2 test environment. This work was demonstrated at the EDCS Demo Days.

Peyman Oreizy continued research on decentralized software evolution. As an evaluation exercise, he componentized a small portion of the Mozilla Web browser -- the open-source version of Netscape's Communicator Web browser -- by adding several software connectors to its architecture. Exposing Mozilla's software architecture to third-party developers allows them to change its functionality in ways that were not previously possible [Ore00].

We have built ArchStudio 2.0, an extensible, integrated software architecture development environment. ArchStudio 2.0 incorporates a number of UCI's software architecture technologies, including: (a) ArchShell, an tool that enables runtime modification of software architectures; (b) Argo/C2, the graphical design environment that uses critics to continuously analyze software designs; (c) DRADEL's parser and topological constraint checker, a toolset that imports and statically analyzes C2 SADEL ADL descriptions; (d) Chimera integration, which allows hypertext linking between our design environment and other software artifacts such as source code and requirements documents; (e) a Web browser for downloading additional software components; and (f) off-the-shelf XML tools for viewing and manipulating the common architectural model. All of these tools use an extensible XML-based architectural model. This permits the addition of new attributes and properties as new tools are added to the environment. The defined XML DTD, called xADL, is designed to support a range of architectural models, including C2 [MOT-00,

KGP-00].

[<http://www.ics.uci.edu/pub/qrch/archstudio>](http://www.ics.uci.edu/pub/qrch/archstudio)

1.2.4. Event-based Architectures

David Rosenblum and Elisabetta Di Nitto (CEFRIEL/Politecnico di Milano) began investigating the problem of reconciling middleware constraints with architectural modeling, with a focus on middleware infrastructures for event-based architectures. Middleware infrastructures are becoming a significant component of large-scale software systems. Traditional software process models encourage postponement of implementation decisions as late in the lifecycle as possible, but this is no longer possible for middleware-based architectures and systems. Architectural models can severely restrict the choice of middleware infrastructures, and middleware infrastructures can impose strong constraints on architectural models. Therefore, languages and environments for architectural modeling must account for the constraints imposed by middleware infrastructures. As a first step in studying this issue, existing ADLs were evaluated as to their suitability for expressing and capturing constraints of event-based styles and event-based infrastructures. This work resulted in a publication for ISAW-3 [CDRW98] and a paper accepted to ICSE'99 [DR99].

Siena is an advanced middleware infrastructure that implements content-based routing to support event-based applications on a wide-area network. The components of a loosely-coupled system are typically designed to operate by generating and responding to asynchronous events. An event notification service is an application-independent infrastructure that supports the construction of event-based systems. The two primary services that should be provided to components by the infrastructure are notification selection and notification delivery. Numerous event notification services have been developed for local-area networks, generally based on a centralized server to select and deliver event notifications. Therefore, they suffer from an inherent inability to scale to wide-area networks, such as the Internet, where the number and physical distribution of the service's clients can quickly overwhelm a centralized solution. The critical challenge in the setting of a wide-area network is to maximize the expressiveness in the selection mechanism without sacrificing scalability in the delivery mechanism. The first version of Siena has been publicly released. The package includes a peer-to-peer server (implemented in C++), a hierarchical server (implemented in Java), and a client-side API for both the C++ and the Java language [CDRW98,CRW00].

1.2.5. C2 and Java Beans

David Rosenblum and his students Rema Natarajan and Doris Tonne have been investigating techniques for supporting architectural concerns in the use of component interoperability standards. Rema Natarajan has developed an enhancement to Sun's Beans Development Kit, called *Arabica*. Arabica's goal is to enable composition of JavaBeans components in the C2 architectural style. The JavaBeans developer expects an interface that enables easy, visual composition of individual JavaBeans into applications that are usable. Arabica not only allows this kind of visual JavaBeans composition, but also enforces C2 stylistic rules and provides customization mechanisms that allow an architect to compose JavaBeans according to the requirements of his or her particular specification of a C2 architecture. This is achieved with a wrapping mechanism that wraps every bean and customizes it into a C2 component. The wrapper also takes care of queuing outgoing events from the component after converting them into C2 messages and queuing incoming messages and converting them into JavaBeans events. Standard C2 connectors are provided, and an

additional mechanism is provided to allow an architect to specify his or her own filtering or monitoring mechanisms and embed them into a C2 connector. The visual composition environment allows the architect to then link the wrapped components and connectors into the C2 architecture. C2 style rules guide this composition task. These features were implemented by taking Sun's Bean Development Kit (BDK 1.0) and extending it to implement the features of our tool. This work resulted in a publication for ISAW-3 [NR98]. Doris Tonne has developed an enhancement to the Beans Development Kit and the definition of the JavaBeans "design pattern", called *Robusta*, which supports the evaluation of methods that incorporate bean invariants and bean method pre- and post-conditions.

1.3. Process

The process work has been directed at producing an open, distributed, extensible process execution environment which leverages WWW protocols and the platform portability afforded by Java. Additionally we have looked to provide effective means of customizing and reusing process fragments, applying the lessons learned by the workflow and computer-supported cooperative work communities, and effective two-way integration with commonly used desktop applications using a variety of programming languages. A prototype system, called Endeavors, serves as an evolving platform for exploring these issues. The Endeavors process support system provides a web-based workflow for distributed team coordination, collaboration, and communication. The system supports lightweight, mobile, graphically specifiable, dynamic, reflexive, componentized, workflow processes that allow execution of workflows across multiple machines and devices by both technical and non-technical users.

1.3.1. Endeavors

The Endeavors process support system generated a great deal of interest in the corporate community. The project was involved in several technology transition efforts:

Users/Companies who have evaluated Endeavors for commercial and other uses:

- Netscape Communication Corporation, Jim Cunningham
- Sony Pictures Imageworks, Jim Berney
- Oracle Corporation, Bahram Saghari
- FileNet Corporation, Bruce Waddington
- MCC, Nancy Perry
- ISI, David Benjamin

Users/Companies who have deployed, plan to, or are currently using Endeavors to deploy workflow processes include:

- Sun Microsystems, Tom Arkwright
- PacBell, Ray Licon
- CoCreate/HP, Mark H. Allen
- Personal Software Process, A. Winsor Brown
- TACOM, Mike Saboe

Under the direction of Ray Licon, a systems analyst at Pacific Bell, we developed a process for the PacBell Applications Development Group (ADG). Licon presented the ADG process using

Endeavors in an executive meeting where he demonstrated how Endeavors creates, executes, and maintains workflow processes.

Sun Microsystems is using Endeavors as a process execution environment for their internal Java education program. An Endeavors process was created for Tom Awkwright to drive Sun's web-based instruction delivery system called JavaBrain. A series of collection forms were then designed and implemented sufficiently to allow a user with little training to develop a complete training course. Java Train can include visual aids such as gif graphics or Java Applets and can create True false, Multiple Choice, and Match Word type "questions". Courses are easily created and presented over the WWW through Endeavors. The Endeavors process provides Sun with an easily extensible means to coordinate multiple courses and collect metrics on the course takers. The system will be used on Sun's new Java Stations. The University has now licensed the JavaBrain technology to Sun Microsystems who is using the JavaBrain technology for the development and deployment of Computer Based Training (CBT) materials.

The University of California, Irvine licensed the Endeavors technology to Endeavors Technology Incorporated (ETI). ETI is a small, highly responsive technology company that will provide Web-based workflow, wireless workflow, Web-based information systems, and E-Business solutions. ETI will use Endeavors to build several products: a lightweight, HTTP-based workflow server that easily integrates with existing tools and technologies at use in a business; a full featured graphical environment for non-technical users to visualize, describe, and share executable descriptions of their work; a lightweight, HTTP-based document management repository; and a scaled down, small footprint version of the Endeavors desktop for use on PDAs, palmtops, and handheld computing devices.

The Course Syllabus Process (CSP) has been deployed at UCI. The CSP lets end users systematically design a course syllabus by following a prescribed process using the Endeavors process engine. If necessary, any task of the CSP can be assigned and routed to different people. Professors and lectures at the Irvine campus now use the CSP for their courses.

Development milestones include the implementation of a distributed infrastructure. Severization of the Endeavors foundation layer provides full distributed processing between remote Endeavors systems as well as support for file locking, remote handler invocation, and distributed process invocation. We improved the Endeavors user interface by incorporating the JDK-1.1 event model and have also added support for the automatic generation of web-based guidance pages. We defined our own Endeavors development and release process and are using this process in our daily development efforts.

Endeavors developers completed a three tier database middle-ware module for the Endeavors system. This system consists of a programmer's interface for the client level (tier three), a middle tier web servlet module (tier two), and a RDBMS (tier one). The client connects to the middle tier using the provided API calls and utility functions from tier one. The middle tier module extends the Java HTTPServlet class and relies on a web server. The middle tier to tier one connection has currently been implemented using the JDBC which makes connections to the MYSQL RDBMS located at tier one. Communication between the tiers is via HTTP and will therefore be considered firewall friendly on many systems. The communication between the client and middle tier is based on ANSI SQL; this enables complex queries, joins, orderings and aggregations. The system supports prepared statements and multiple databases. This database middle-ware is compatible

with the Endeavors user level Handler model. This feature provides a valuable tool for Endeavors users that require data storage into proprietary systems such as Oracle, MSSQL, and Sybase. In fact, handlers can be written to maintain user data into any DBMS that has an available JDBC driver. This model enables Endeavors to interact with and maintain data and processes which rely on data from legacy data stores.

Endeavors has been connected to Oracle8 ORDBMS as a test-bed for transaction management within a database supported process environment. This will allow us to investigate the benefits of a transactional workflow system and to describe an optimal transaction manager model for the process environment.

An initial implementation of the Endeavors User and System levels as C2 components was completed. Using the C2-style allows the System and User levels bi-directional communication through any of the C2 Connector objects. The C2-style enables various Endeavors architectures to be constructed. For example, a server side architecture is created with a standard C2 connector welded to the System level component and the other end of the connector accessible by a servlet. The servlet is part of a C2 connector which uses a web server and a combination of Java's Socket, ServerSocket, and URL Connection classes to form the connector. This allows a User level component running as either an applet or as a stand alone application to connect to a remote System level component. Thus, multiple distributed User level clients can connect to the same System level component and receive updates when changes are made. Ninety-six of the messages have been implemented and we have validated the design.

Two prototype workflow analysis tools have been developed. The first tool converts control flow based workflows into a Petri-net. The conversion requires that the control flow and all of its artifacts are converted into a data-flow model. By feeding the data-flow model into the Petri-net tool we can analyze five important attributes for workflow: Soundness of workflow, syntax, time analysis, simulations, and axiomatic verification. Soundness of workflow checks for reachability and completion, deadlock and livelock. Deadlock detection algorithms assure that one or more processes will not reach starvation due to conflicting or unavailable resources. Reachability analysis will ensure that the workflow will always start and end properly. Time analysis can be used to calculate minimum, average and maximum time for completion of a workflow. Simulations can provide cognitive walkthroughs to users before a system is deployed. Axiomatic verification can verify properties pertaining to a workflow.

The second tool called RealityCheck, supports analysis and filtering of data generated from adaptive workflow management tools. RealityCheck compares the history of completed workflow processes against the initially prescribed workflow. After the comparative analysis, RealityCheck constructs a new generalized process using a smoothing algorithm. The process manager can then interactively customize the new process by manipulating different aspects of the process.

Several reusable process components were created to support rapid development and reuse of workflow processes. A library of components, called e-lib (Endeavors libraries), was developed for the domain of document routing and approval, and web-based workflows. The web-based workflow components allow end users to easily assign web pages as work activities. By using these web-based workflow components and WebNavigator, an interface that lies between an HTTP server and Endeavors, end users can initiate, continue complete and change a workflow process from a URL or HTML form.

The Endeavors group has been researching integration strategies with the latest event based architectures and tools. Recent CSCW tools (used to support the end users for work) provide rich event based communications and can be used as the integration mechanism for workflow systems. Endeavors is continuing to leverage and study the semantics of these event mechanisms for discovering and integrating process into everyday tools. This approach, consistent with the Endeavors design philosophy, lowers the cost of adoption and through Endeavors allows for better automation and analysis of work. New web-based reusable components as well as refinements to existing components have been made to the Endeavors web-based process library, e-lib. These tools allow process managers to easily assign web pages as work activities and provide seamless integration of the WWW activities and resources with process.

Endeavors has designed and implemented a set of new and reusable process components for the field of scheduling and reservation systems. This process enables end users to add, modify, and remove their schedules from a database and uses the Endeavors process system to manage the user interactions and transactions as a high level processes. Typically, these systems interact with database systems which model these as transactions and enforce them through a database transaction manager to ensure their consistency. However, as the resource reaches capacity, the point at which exceptions occur, transaction systems are often too rigid to deal with these problems gracefully. By using the high level and interactive process components, dynamic flexibility and exception handling capabilities of Endeavors, Endeavors better resolves these complex scheduling issues, especially those that require coordination such as group calendaring systems.

[<http://www.ics.uci.edu/pub/endeavors/>](http://www.ics.uci.edu/pub/endeavors/)

1.3.2. EDEM

Expectation-Driven Event Monitoring EDEM provides developers with a platform for creating software agents to collect usability data and increase user involvement in the evolution of interactive systems. EDEM collects information that is currently lost regarding actual usage of applications to promote a more empirically grounded design process. The Expectation-Driven Event Monitoring (EDEM) substrate enables software engineers to define agents to monitor the usability of Java applications and evolve systems based on the resulting feedback.

EDEM was redesigned for compliance with the Java 1.1 event model. Features include a flexible and efficient event dispatch mechanism, enhanced data collection and reporting options, an API to allow input and output of arbitrary events for monitoring, and default expectation agents with wizards to aid in expectation specification and agent parameterization. In addition, a survey of automated techniques for extracting usage/usability information from automatically captured user interaction events was completed [Hil98].

EDEM was successfully integrated by Lockheed Martin C2 Integration Systems into the Global Transportation Network (GTN) demonstration scenario for the Third Annual EDCS Demo Days in Baltimore in July.

David Hilbert managed a development effort at Microsoft Corporation in which an application (with over 1000 commands and 300 dialogs) was instrumented to collect usage data regarding the behavior of 500-1000 users using the application over a two month period. This industrial collaboration was initiated to allow the principles and techniques underlying UCI's EDEM research (in the area of large-scale collection of application usage data) to be evaluated within the context of a

large-scale industrial project.

[<http://www.ics.uci.edu/~dhillbert/edem/>](http://www.ics.uci.edu/~dhillbert/edem/)

1.3.3. Knowledge Depot

Knowledge Depot is a tool for supporting Project Awareness. Originally a group memory used to store and automatically organize project related email and documents, it has evolved into a project awareness tool which allows geographically distributed project members to subscribe to specific categories of information in the group memory. These subscriptions cause the system to send users summaries of all new information to arrive in that subject, thus allowing people to remain aware of discussions and documents on project topics that affect them.

Development of the system has included enhancing Java Knowledge Depot's usability in response to initial user feedback, enhancing Bell Atlantic's Lotus Notes subscription feature to scale up to handling the thousands of people who use the Lotus Notes Knowledge Depots. Knowledge Depot has gained a new web based interface that has resulted in significant gains in usability over previous interfaces. It has also gained noticeably in speed. These two factors have made the system far more approachable to users.

The scalability of Knowledge Depot was enhanced to support greater numbers and types of users. The new design also allows greater flexibility in specifying relevant information, as well as in clustering the information together within a single notification. This will enable studies within the Open Source community and with user groups on the University campus in determining the exact benefits and potential of this system.

2. Professional Personnel

Faculty:

David Redmiles
David S. Rosenblum
Richard N. Taylor

Research Assistants:

Ken Anderson
Gregory Bolcer
Roy Fielding
David Hilbert
Arthur Hitomi
Peter Kammer
Michael Kantor
Rohit Khare
Neno Medvidovic
Rema Natarajan
Peyman Oreizy
Jason Robbins
Shilpa Shukla
James Whitehead

Research Programmers:

Clay Cover
Adam Gauthier
Yuzo Kanomata
Edwin Kraemer
Kari Nies

Visiting Scholars:

Elisabetta Di Nitto, *Politecnico di Milano, Italy*
Alfonso Fuggetta, *Politecnico di Milano, Italy*

Degrees Awarded:

Peyman Oreizy, Ph.D. Computer Science

Awarded January 2000

Dissertation: "*Open Architecture Software: A Flexible Approach to Decentralized Software Evolution*" [Ore00]

Jason Robbins, Ph.D. Computer Science

Awarded September 1999

Dissertation: "*Cognitive Support Features for Software Development Tools*" [Rob99]

David Hilbert, Ph.D. Computer Science

Awarded June 1999

Dissertation: "*Large-Scale Collection of Application Usage Data and User Feedback to Inform Interactive Software Development*" [Hil99]

Nenad Medvidovic, Ph.D. Computer Science

Awarded December 1998

Dissertation: "*Architecture-based Specification-time Software Evolution*" [Med98]

Gregory Alan Bolcer, Ph.D. Computer Science

Awarded November 1998

Dissertation: "*Flexible and Customizable Workflow Execution on the WWW*" [Bol98]

Kenneth M. Anderson, Ph.D.

Awarded June 1997

Dissertation: "*Pervasive Hypermedia*" [And97-3]

3. Publications in Technical Journals

Kenneth M. Anderson, Richard N. Taylor, and E. James Whitehead, Jr. *A Critique of the Open Hypermedia Protocol*. Journal of Digital Information, special issue on open hypermedia systems, vol. 1, issue 2. [ATW97]

Gregory Alan Bolcer and Richard N. Taylor. *Advanced Workflow Management Technologies*. Journal of Software Process Improvement and Practice, vol. 4, no. 3, 1998, pp. 125-171. [BT98]

Roy Fielding, E. James Whitehead, Jr., Kenneth Anderson, Peyman Oreizy, Gregory Bolcer, and Richard Taylor. *Web-based Development of Complex Information Products*. Communications of the ACM, vol. 41, no. 8, pp. 84-92. August 1998. [FWA+98]

Peter Kammer, Greg Bolcer, Richard N. Taylor, Arthur Hitomi, and Mark Bergman. *Techniques for Supporting Dynamic and Adaptive Workflow*. Journal of Computer Supported Cooperative Work, vol. 9, no. 3, November 2000, pp. 269-292. [KBT+00]

Nenad Medvidovic, Richard N. Taylor. *Exploiting Architectural Style to Develop a Family of Application*. IEE Proceedings Software Engineering journal, Oct/Dec 1997. [MT96-2]

Nenad Medvidovic and Richard N. Taylor. *A Classification and Comparison Framework for Software Architecture Description Languages*. IEEE Transactions on Software Engineering, vol. 26, no. 1, January 2000. [MT99]

Peyman Oreizy, Michael M. Gorlick, Richard N. Taylor, Dennis Heimbigner, Gregory Johnson, Nenad Medvidovic, Alex Quilici, David S. Rosenblum, and Alexander L. Wolf. *An Architecture-Based Approach to Self-Adaptive Software*. IEEE Intelligent Systems, vol. 14, no. 3, May/June 1999, pp. 54-62. [OG+99]

Peyman Oreizy, Richard N. Taylor. *On the Role of Software Architectures in Runtime System Reconfiguration*. IEE Transactions Software Engineering, vol. 145, no. 5, October 1998. [OT98-2]

Jason E. Robbins, David M. Hilbert, and David F. Redmiles. *Extending Design Environments to Software Architecture Design*. Automated Software Engineering, vol. 5, no. 3. pp. 261-290, July 1998. [RHR98-2]

Jason Robbins and David Redmiles. *Software Architecture Critics in the Argo Design Environment*. Journal of Knowledge-Based Systems, vol. 11, no. 1., pp. 47-60, September 1998. [RR98]

Jason Robbins and David Redmiles. *Cognitive support, UML Adherence, and XMI Interchange in Argo/UML*. Journal of Information and Software Technology, special issue on the best of CoSET 99, vol 42, 2000, pp. 79-89. [RR99-2]

E. James Whitehead. *World Wide Web Distributed Authoring and Versioning (WebDAV): An Introduction*. ACM StandardView, March, 1997. [Whi97]

4. Final Resource Status Report

Table 1: Contract Resource Status: 11/01/96 - 04/30/200

Categories	Total Funds Rec'd	Expenses To Date	Balance of Funds Rec'd
Salaries & Wages	1,037,477	1,128,762	-91,285
Fringe Benefits	102,492	147,168	-44,676
Materials & Supplies	44,848	53,780	-8,932
Travel	181,436	212,985	-31,549
Other	71,944	0	71,944
Indirect Costs	765,656	761,482	4,174
Equipment	177,960	177,950	10
Tuition & Fees	175,822	72,250	103,572
Total	2,557,635	2,554,377	3,258

References

- [And97] Kenneth M. Anderson. *Integrating Open Hypermedia Systems with the World Wide Web*. Proceedings of Hypertext'97, the Eighth ACM Conference on Hypermedia Technology, Southampton, UK, April 6-11, 1997.
- [And97-2] Kenneth M. Anderson. *A Critique of the Open Hypermedia Protocol*. Proceedings of the Third Workshop on Open Hypermedia Systems, held in conjunction with Hypertext'97, Southampton, UK, April 6-7, 1997.
- [And97-3] Kenneth M. Anderson. *Pervasive Hypermedia*. Ph.D. Dissertation, Department of Information and Computer Science, University of California, Irvine, June 1997.
- [And98] Kenneth M. Anderson. *Client-Side Services for Open Hypermedia: Getting past the 'foo'...*. Proceedings of the 4th International Workshop on Open Hypermedia Systems held at Hypertext'98. Pittsburgh, Pennsylvania, June 20-24, 1998.
- [And99] Kenneth M. Anderson. *Data Scalability in Open Hypermedia Systems*. In Proceedings of the ACM Hypertext '99 Conference, Darmstadt, Germany, February 21-25, 1999.
- [And99-2] Kenneth M. Anderson. *Supporting Industrial Hyperwebs: Lessons in Scalability*. In Proceedings of the 1999 International Conference on Software Engineering, Los Angeles, CA, May 16-22, 1999.
- [ATW97] Kenneth M. Anderson, Richard N. Taylor, and E. James Whitehead, Jr. *A Critique of the Open Hypermedia Protocol*. Journal of Digital Information, special issue on open hypermedia systems, vol. 1, no. 2.
- [Bol96] Gregory Alan Bolcer. *Endeavors: A Process System Integration Infrastructure*. Proceedings of the International Conference on Software Process. Brighton, UK, December 1996.
- [BFM98] Tim Berners-Lee, Roy T. Fielding, and Larry Masinter. *Uniform Resource Identifiers (URI): Generic Syntax*. Internet RFC 2396, August 1998.
- [Bol98] Gregory Alan Bolcer. *Flexible and Customizable Workflow Execution on the WWW*. Ph.D. Dissertation, Department of Information and Computer Science, University of California, Irvine, November 1998.
- [BT98] Gregory Alan Bolcer and Richard N. Taylor. *Advanced Workflow Management Technologies*. The Journal of Software Process Improvement and Practice, vol. 4, no. 3, 1998, 125-171.
- [BK99] Gregory Alan Bolcer and Gail Kaiser. *SWAP: Leveraging the Web to Manage Workflow*. IEEE Internet Computing, vol. 3, no. 1, Jan-Feb, 1999.
- [CDRW98] A. Carzaniga, E. Di Nitto, D.S. Rosenblum and A.L. Wolf, *Issues in Supporting Event-based Architectural Styles*. In Proceedings of the Third International Software Architecture Workshop, Lake Buena Vista, FL, Nov. 1998.
- [CRW00] Antonio Carzaniga, David S. Rosenblum, Alexander L. Wolf. *Achieving Scalability and Expressiveness in an Internet-Scale Event Notification Service*. To appear in Proceedings of the Nineteenth ACM Symposium on Principles of Distributed

Computing (PODC 2000), July 16-19 2000, Portland, Oregon.

- [DMT99] Eric M. Dashofy, Nenad Medvidovic, and Richard N. Taylor. *Using Off-The-Shelf Middleware to Implement Connectors in Distributed Software Architectures*. To appear in Proceedings of the 1999 International Conference on Software Engineering, Los Angeles, CA, May 16-22, 1999.
- [DR99] Elisabetta Di Nitto and David Rosenblum. *Exploiting ADLs to Specify Architectural Styles Induced by Middleware Infrastructures*. To appear in Proceedings of the 1999 International Conference on Software Engineering, Los Angeles, CA, May 16-22, 1999.
- [Dud98] Andrea Dudrow. *WebDAV to tackle team authoring*. MacWeek On-line: vol 12, issue 17; May 7, 1998. <http://macweek.zdnet.com/mw_1218/in_webdav.html>
- [FGM+99] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. *Hypertext Transfer Protocol -- HTTP/1.1*. Internet Draft Standard Request for Comments 2616. June 1999.
- [Fie97] Roy T. Fielding. *WebSoft: Building a Global Software Engineering Environment*. Presented at the ICSE 97 Workshop on Software Engineering and the World Wide Web. Boston, May 19, 1997. <http://ricis.cl.uh.edu/SEWWW/Submissions/Fielding/>
- [FT00] Roy T. Fielding and Richard N. Taylor. *Principled Design of the Modern Web Architecture*. To appear in Proceedings of the International Conference on Software Engineering (ICSE 2000), Limerick, Ireland, June 4-11, 2000.
- [FWA+98] Roy Fielding, E. James Whitehead, Jr., Kenneth Anderson, Peyman Oreizy, Gregory Bolcer, and Richard Taylor. *Web-based Development of Complex Information Products*. Communications of the ACM, vol. 41, no. 8, pp. 84-92. August 1998.
- [GWF+99] Y. Goland, E. Whitehead, A. Faizi, S. Carter, and D. Jensen. *HTTP Extensions for Distributed Authoring -- WEBDAV*. Internet Proposed Standard Request for Comments 2518. February, 1999.
- [Han98] Brian Hannon. *WebDAV to ease Web authoring*. PC Week On-line, June 22, 1998. <<http://www.zdnet.com/pcweek/news/0622/22webdav.html>>
- [Hil98] David M. Hilbert. *A Survey of Computer-Aided Techniques for Extracting Usability Information from User Interface Events*. Technical Report UCI-ICS-98-13, Department of Information and Computer Science, University of California, Irvine, March 1998.
- [Hil99] David M. Hilbert. *Large-Scale Collection of Application Usage Data and User Feedback to inform Interactive Software Development*. Ph.D. Dissertation, Department of Information and Computer Science, University of California, Irvine, June 8, 1999.
- [HR98] David M. Hilbert, David F. Redmiles. *An Approach to Large-Scale Collection of Application Usage Data Over the Internet*. In Proceedings of the 1998 International Conference on Software Engineering, Kyoto, Japan, April 19-25, 1998.
- [HR98-2] David M. Hilbert, David F. Redmiles. *Agents for Collecting Application Usage Data*

Over the Internet. In Proceedings of the 1998 Autonomous Agents Conference (Agents'98), Minneapolis/St. Paul, May 9-13, 1998.

- [HR98-3] David Hilbert, David Redmiles. *Separating the Wheat from the Chaff in Internet-Mediated User Feedback*. In Proceedings of the Workshop on Internet-based Groupware for User Participation in Product Development, 1998 ACM Conference on Computer Supported Cooperative Work (CSCW'98), Seattle, WA, USA, November 14-18, 1998.
- [HRR97] David M. Hilbert, Jason E. Robbins, and David F. Redmiles. *Supporting Ongoing User Involvement in Development via Expectation-Driven Event Monitoring*. Technical Report UCI-ICS-97-19, Department of Information and Computer Science, University of California, Irvine, May 1997.
- [HRR98] David M. Hilbert, Jason E. Robbins, David F. Redmiles. *EDEM: Intelligent Agents for Collecting Usage Data and Increasing User Involvement in Development*. Proceedings of the 1998 Conference on Intelligent User Interfaces, San Francisco, CA, January 1998.
- [HBT97] Arthur S. Hitomi, Gregory Alan Bolcer, and Richard N. Taylor. *Endeavors: A process Sytem Infrastructure*. Proceedings of the 1997 International Conference on Software Engineering, Boston, May 17-23, 1997, pp. 598-599.
- [HKBT98] Arthur S. Hitomi, Peter J. Kammer, Gregory Alan Bolcer, and Richard N. Taylor. *Distributed Workflow using HTTP: Example using Software Pre-requirements*. Demonstration paper. In Proceedings of the 1998 International Conference on Software Engineering, Kyoto, Japan, April 19-25, 1998.
- [KW97] Gail Kaiser, E. James Whitehead. *Collaborative work: Distributed authoring and versioning*. IEEE Internet Computing, Vol. 1, No. 2, March/April, 1997, pp. 76-77.
- [KBB98] Peter Kammer, Greg Bolcer, and Mark Bergman. *Requirements for Supporting Dynamic and Adaptive Workflow on the WWW*. In Proceedings of the CSCW Workshop on Adaptive Workflow Systems, 1998 ACM Conference on Computer Supported Cooperative Work (CSCW'98), Seattle, WA, USA, November 14-18, 1998.
- [KBT+00] Peter Kammer, Greg Bolcer, Richard N. Taylor, Arthur Hitomi, and Mark Bergman. *Techniques for Supporting Dynamic and Adaptive Workflow*. Journal of Computer Supported Cooperative Work, vol. 9, no. 3, November 2000, pp. 269-292.
- [KBTH98] Peter J. Kammer, Gregory Alan Bolcer, Richard N. Taylor, and Arthur S. Hitomi. *Supporting Distributed Workflow Using HTTP*. In Proceedings of the 5th International Conference on Software Process (ICSP5): Computer Supported Organizational Work, Chicago, Illinois, June 14-17, 1998.
- [KGP+00] Rohit Khare, Michael Guntersdorfer, Peyman Oreizy, Nenad Medvivovic, and Richard N. Taylor. *xADL: Enabling Architecture-Centric Tool Integration With XML*. Technical Report UCI-ICS-00-12, Department of Information and Computer Science, University of California, Irvine, April 2000.
- [Kha98] Rohit Khare. *Life in a TP*. IEEE Internet Computing, March/April 1998.

- [Kha98-2] Rohit Khare. *The News about Jon: Reflections on the Wizard of TPs; and other Network News*. IEEE Internet Computing, vol. 3, no. 5, September 1998.
- [Kha98-3] Rohit Khare. *Who Killed Gopher? An Extensible Murder Mystery*. IEEE Internet Computing, vol. 3, no. 6, November 1998.
- [KR98] Rohit Khare and Adam Rifkin. *Composing Active Proxies to Extend the Web*. Proceedings of the OMG-DARPA-MCC Workshop on Compositional Software Architectures.
- [KR98-2] Rohit Khare with Adam Rifkin. *Modeling Data and Metadata: XML and RDF*. Tutorial In Proceedings of 1998 ACM Conference on Computer Supported Cooperative Work (CSCW'98), Seattle, WA, USA, November 14-18, 1998.
- [MDT98] Nenad Medvidovic, Eric M. Dashofy, and Richard N. Taylor. *Employing Off-the-Shelf Connector Technologies in C2-Style Architectures*. In Proceedings of the California Software Symposium (CSS'98), Irvine, CA, pp. 21-30, October 23, 1998.
- [Med97] Nenad Medvidovic. *A classification and comparison framework for software architecture description languages*. UCI technical report (UCI-ICS-97-02).
- [Med98] Nenad Medvidovic. *Architecture-based Specification-time Software Evolution*. Ph.D. Dissertation, Department of Information and Computer Science, University of California, Irvine, November 1998.
- [MOT97] Nenad Medvidovic, Peyman Oreizy, Richard N. Taylor. *Reuse of Off-the-Shelf Components in C2-Style Architectures*. The 1997 Symposium on Software Reusability (SSR'97), Boston MA, May 1997.
- [MOT+00] Nenad Medvidovic, Peyman Oreizy, Richard N. Taylor, Rohit Khare, and Michael Guntersdorfer. *An Architecture-Centered Approach to Software Environment Integration*. Technical Report, UCI-ICS-00-11, Department of Information and Computer Science, University of California, Irvine, March 2000.
- [MR97] Nenad Medvidovic and David S. Rosenblum. *Domains of Concern in Software Architectures and Architecture Description Languages*. Proceedings of the 1997 USENIX Conference on Domain-Specific Languages, Santa Barbara, CA.
- [MR99] Nenad Medvidovic and David S. Rosenblum. *Assessing the Suitability of a Standard Design Method for Modeling Software Architectures*. In Proceedings of the First Working IFIP Conference on Software Architecture (WICSA1), San Antonio, TX, February 22-24, 1999.
- [MRT98] Nenad Medvidovic, David S. Rosenblum, and Richard N. Taylor. *A Type Theory for Software Architectures*. Technical Report, UCI-ICS-98-14, Department of Information and Computer Science, University of California, Irvine, April 1998.
- [MRT99] Nenad Medvidovic, David S. Rosenblum, and Richard N. Taylor. *A Language and Environment for Architecture-Based Software Development and Evolution*. In Proceedings of the 21st International Conference on Software Engineering (ICSE'99), Los Angeles, CA, May 16-22, 1999.
- [MT96] Nenad Medvidovic, Richard N. Taylor. *Reusing Off-the-shelf Components to*

Develop a Family of Applications in the C2 Architectural Style. Presented at the International Workshop on Development and Evolution of Software Architectures for Product Families, Las Navas del Marquis, Avila, Spain, November 18-19, 1996.

- [MT96-2] Nenad Medvidovic, Richard N. Taylor. *Exploiting Architectural Style to Develop a Family of Application*. IEE Proceedings Software Engineering journal, Oct/Dec 1997.
- [MT97] Nenad Medvidovic, Richard N. Taylor. *A Framework for Classifying and Comparing Architecture Description Languages*. Proceedings of the Sixth European Software Engineering Conference, held in conjunction with the Fifth ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE). September 22-25, 1997.
- [MT98] Nenad Medvidovic and Richard N. Taylor. *Separating Fact from Fiction in Software Architecture*. In Proceedings of the Third International Software Architecture Workshop (ISAW-3), pages 105-108, Orlando, FL, November 1-2, 1998.
- [MT99] Nenad Medvidovic and Richard N. Taylor. *A Classification and Comparison Framework for Software Architecture Description Languages*. IEEE Transactions on Software Engineering, vol. 26, no. 1, January 2000. [MT99]
- [MTR98] Nenad Medvidovic, Richard N. Taylor, and David S. Rosenblum. *An Architecture-Based Approach to Software Evolution*. In Proceedings of the International Workshop on the Principles of Software Evolution (IWPSE 98), Kyoto, Japan, April 20 - 21, 1998.
- [NR98] Rema Natarajan and David S. Rosenblum. *Merging Component Models and Architectural Styles*. In Proceedings of the Third International Software Architecture Workshop (ISAW-3), Orlando, FL, November 1-2, 1998.
- [OGT+99] Peyman Oreizy, Michael M. Gorlick, Richard N. Taylor, Dennis Heimbigner, Gregory Johnson, Nenad Medvidovic, Alex Quilici, David S. Rosenblum, and Alexander L. Wolf. *An Architecture-Based Approach to Self-Adaptive Software*. IEEE Intelligent Systems, vol. 14, no. 3, May/June 1999, pp. 54-62.
- [OMT98] Peyman Oreizy, Nenad Medvidovic, Richard N. Taylor. *Architecture-based Runtime Software Evolution*. In Proceedings of the 1998 International Conference on Software Engineering, Kyoto, Japan, April 19-25, 1998.
- [OMTR98] Peyman Oreizy, Nenad Medvidovic, Richard N. Taylor and David S. Rosenblum. *Software Architecture and Component Technologies: Bridging the Gap*. Proceedings of the OMG-DARPA-MCC Workshop on Compositional Software Architectures, Monterey, CA, January 1998.
- [Ore97] Peyman Oreizy. *The WWW as an Enabling Technology for Software Engineering*. Presented at the ICSE 97 Workshop on Software Engineering and the World Wide Web. Boston, May 19, 1997. <http://ricis.cl.uh.edu/SEWWW/Submissions/Oreizy/>
- [Ore98] Peyman Oreizy. *Decentralized Software Evolution*. In Proceedings of the International Workshop on the Principles of Software Evolution (IWPSE 98), Kyoto, Japan, April 20 - 21, 1998.

- [Rob98] Jason Robbins. *Design Critiquing Systems*. UCI-ICS Technical Report 98-41, Department of Information and Computer Science, University of California, Irvine, November 1998.
- [RR98] Jason Robbins and David Redmiles. *Software Architecture Critics in the Argo Design Environment*. Journal of Knowledge-Based Systems. vol. 11, no. 1. September 1998. pp. 47-60.
- [RR99] Jason Robbins and David Redmiles. *Cognitive support, UML Adherence, and XMI Interchange in Argo/UML*. In Proceedings of the 1999 Symposium on the Construction of Software Engineering Tools (CoSET'99), Los Angeles, CA, May 17-18, 1999.
- [RR99-2] Jason Robbins and David Redmiles. *Cognitive support, UML Adherence, and XMI Interchange in Argo/UML*. Journal of Information and Software Technology, special issue on the best of CoSET'99, vol 42, 2000, pp. 79-89.
- [RRR97] Jason E. Robbins, David F. Redmiles, and David S. Rosenblum. *Integrating C2 with the Unified Modeling Language*. Proceedings of the 1997 California Software Symposium, Irvine, CA.
- [Ros98] D.S. Rosenblum, *Challenges in Exploiting Architectural Models in Software Testing*, Proc. NSF/CNR Workshop on the Role of Software Architecture in Testing and Analysis, Marsala, Sicily, Jul. 1998, pp. 49-53.
- [Ros98-2] David S. Rosenblum. *Adequate Testing of Component-Based Software*. Department of Information and Computer Science, University of California, Irvine, Technical Report UCI-ICS-97-34, August 1997.
- [SNR98] Shilpa V. Shukla, Bonnie A. Nardi, and David F. Redmiles. *Beta Blockers and Hot Potatoes: Redefining Artifacts in Sociotechnical Design*. Technical Report UCI-ICS-97-45. Department of Information and Computer Science, University of California, Irvine, September 1997.
- [SVWD97] J.A. Slein, F. Vitali, E.J. Whitehead, Jr., D.G. Durand. *Requirements for distributed authoring and versioning on the World Wide Web*. Internet-Draft, work-in-progress, <draft-ietf-webdav-requirements-03.txt>. <ftp://ds.internic.net/internet-drafts/draft-ietf-webdav-requirements-03.txt>
- [TMA+96] Richard N. Taylor, Nenad Medvidovic, Kenneth M. Anderson, E. James Whitehead Jr., Jason E. Robbins, Kari A. Nies, Peyman Oreizy, and Deborah L. Dubrow. *A Component- and Message-Based Architectural Style for GUI Software*. IEEE Transactions on Software Engineering, Vol. 22, No. 6, June 1996, pp. 390-406.
- [VRR99] Jaya Vaidyanathan, Jason Robbins, and David Redmiles. *Using HTML to Create Early Prototypes*. In Proceedings of the 1999 Conference on Human Factors in Computing Systems(CHI'99), Pittsburgh, PA, May 15-20, 1999.
- [Wal98] Jeff Walsh. *Web-DAV protocol to simplify Web authoring*. InfoWorld, April 28, 1998.
- [WebD97-4] WebDAV Design Team. *Distributed Authoring Scenarios*. WebDAV requirements document released by the Internet Engineering Task Force as an Internet-Draft. May

31, 1997.

- [WG99] E. James Whitehead, Jr. and Yaron Y. Goland. *WebDAV: A network protocol for remote collaborative authoring on the Web*. In Proceedings of the European Conference of Computer Supported Cooperative Work (ECSCW'99), Copenhagen, Denmark, September 12-16, 1999, pp. 291-310.
- [Whi97] E. James Whitehead. *World Wide Web Distributed Authoring and Versioning (WebDAV): An Introduction*. ACM StandardView, March, 1997.
- [Whi97-2] E. James Whitehead, Jr. *An Architectural Model for Application Integration in Open Hypermedia Environments*. Proceedings of Hypertext'97, the Eighth ACM Conference on Hypermedia Technology, Southampton, UK, April 6-11, 1997.
- [Whi98] E. James Whitehead, Jr. *Lessons from WebDAV for the Next Generation Web Infrastructure*. Presented at Towards a New Generation of HTTP, A workshop on global hypermedia infrastructure, held with the 7th International World Wide Web Conference, Brisbane, Australia, QLD, April 14, 1998.
- [Whi99] E. James Whitehead, Jr. *Control Choices and Network Effects in Hypertext Systems*. In Proceedings of the ACM Hypertext '99 Conference, Darmstadt, Germany, February 21-25, 1999, pp. 75-82.
- [Whi99-2] E. James Whitehead, Jr. *Goals for a Configuration Management Network Protocol*. In Proceedings of the Ninth International Symposium on System Configuration Management (SCM-9), LNCS 1675, Toulouse, France on September 5-7, 1999, pp. 186-204.
- [WM98] E. J. Whitehead, Jr. and M. Murata. *XML Media Types*. U.C. Irvine, Fuji Xerox, Internet-Draft, work-in-progress, draft-whitehead-xml-mime-05, June 26, 1998. <<ftp://ftp.isi.edu/in-notes/rfc2376.txt>>
- [WW97] Uffe Kock Wiil and E. James Whitehead, Jr. *Interoperability and Open Hypermedia Systems*. Proceedings of the Third Workshop on Open Hypermedia Systems, held in conjunction with Hypertext'97, Southampton, UK, April 6-7, 1997.

**MISSION
OF
AFRL/INFORMATION DIRECTORATE (IF)**

*The advancement and application of Information Systems Science
and Technology to meet Air Force unique requirements for
Information Dominance and its transition to aerospace systems to
meet Air Force needs.*